

This Application Note is pertinent to the Unidrive SP, Commander GP20 Affinity and Commander SK using SyPT Lite

Creating a Coded Flashing Indicator Light

Introduction

On occasion one may have machine conditions that would indicate some kind of problem or something that needs taken care of. For example, a machine might have :

- Guard Not in Place
- Machine Not Up to Temperature
- Temperature of something Too High
- Lube Pump Not On
- Jam in Entry Section

Obviously, these are things that are easily displayed on an HMI (Keypad/Display Unit) or could be indicated with a simple light for each condition. However, for low cost/space reasons, you may be called upon to provide a simple Flashing light that will flash out a code if one or more conditions exist. This application note will illustrate how one can create a simple coded Flasher/Pulser as described.

This application note will use the Free software SyPT Lite that can be utilized by our AC Drive families.

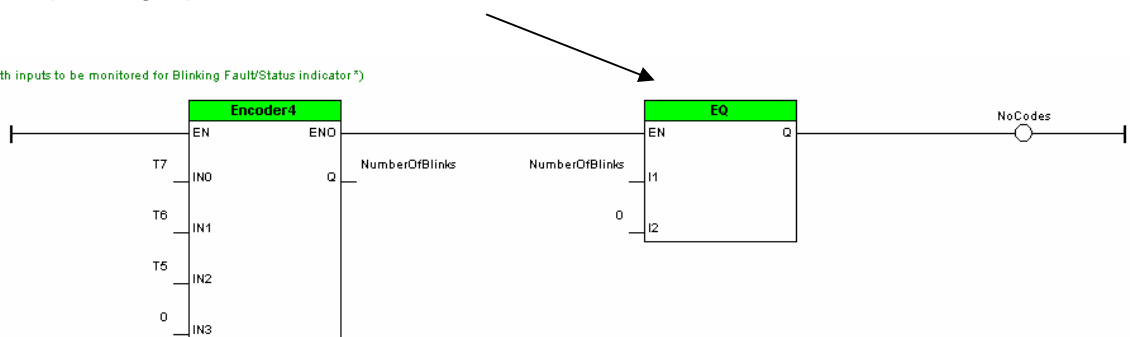
To obtain SyPT Lite click here→ [SyPT Lite](#)



Implementation

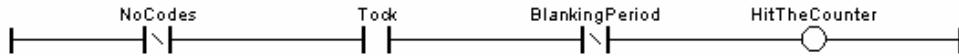
We could handle more conditions but this implementation only considered 4 external conditions. In order to encode those individual conditions we could use the Encoder function block provided by the SyPT Lite library. This function block will create a decimal number based on the binary value present on its inputs. The decimal number will become the number of blinks or flashes that we wish to display. We need to disable the flashing indicator when there are no pending inputs to annunciate hence the test for 0.

(* Feed encoder with inputs to be monitored for Blinking Fault/Status indicator*)



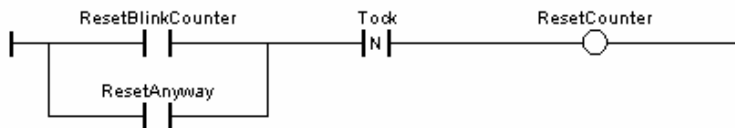
The logic below provides a “blinking period” between the flashing light sequences so that one can synchronize with the flasher and begin counting. Tock is a one second flasher as created and illustrated in [CTSL001](#).

(To provide a blanking period between blink code display)

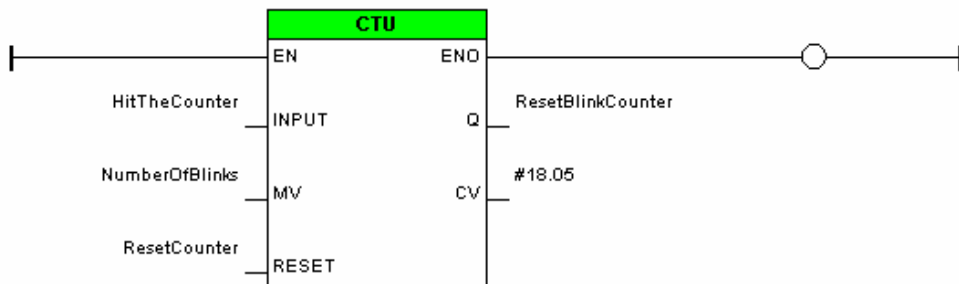


The next rung of logic will reset the Number of Blinks counter after that code has been flashed.

(To ensure last blink is full duration)



(Double-click to enter a rung comment)

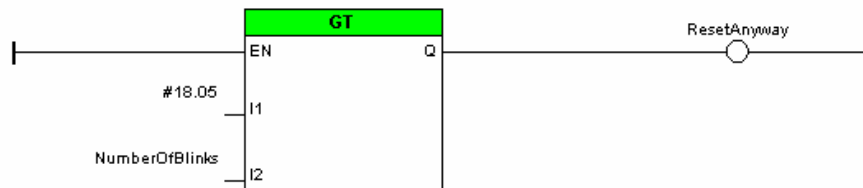


Special consideration was given to account for a couple exceptional conditions.

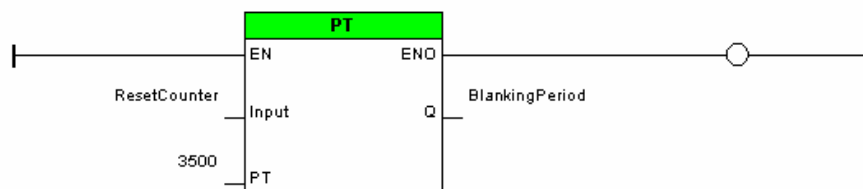
Firstly, one would desire to see complete flashes and not a partial flash at either the beginning or end of a flashed out sequence. This was taken care of by creating a secondary counter “resetter” synchronized to the negative edge of the Seconds Tick.

Secondly, should the coded number change from say 6 to 4 while or after the flasher began flashing out its code, we should reset the counter anyway to start over with a new accurate sequence.

(If the number changes to a lesser value during the code flashing sequence, Reset and start over)



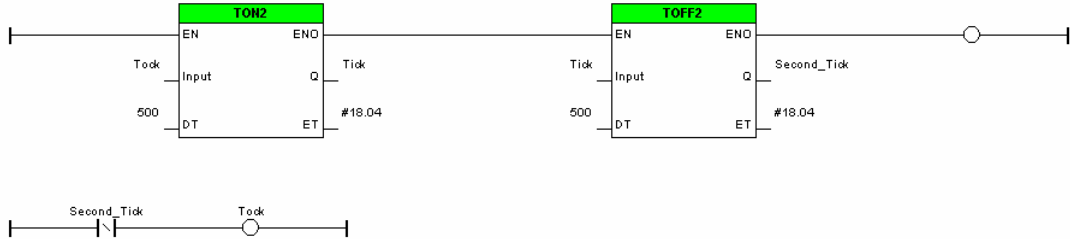
(When the complete sequence has been flashed out, create a dead time space)



Seconds pulser/generator

See [CTSL001](#) for details

(* One Half of Astable Multivibrator *)



(* *)

Register Assignments and Other Notes

Used 1 drive register and assigned individual bits to save memory space

#18.04, 5 and 6 are just temporary internal values used by this program

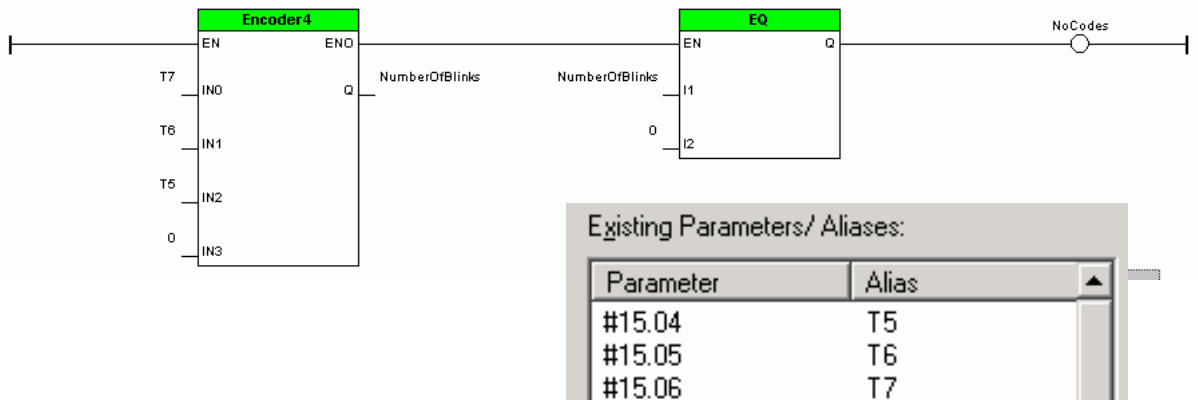
This is what you would assign to a digital output or relay to drive a flashing coded light

| Parameter | Alias |
|-----------|-------------------|
| #18.02.01 | NoCodes |
| #18.02.03 | ResetCounter |
| #18.02.04 | ResetBlinkCounter |
| #18.02.05 | BlankingPeriod |
| #18.02.06 | ResetAnyway |
| #18.02.07 | Tick |
| #18.02.08 | Tock |
| #18.02.09 | Second_Tick |
| #18.04 | |
| #18.05 | |
| #18.06 | NumberOfBlinks |
| #18.50 | HitTheCounter |

Operation and other Details

In this particular example, I was using a Commander SK (specifically one of our SK Training units). Since this Training unit had an I/O Lite module included for extra I/O capability, I decided to use the digital inputs provided on that unit to simulate external conditions I wish to indicate. There are switches on this trainer wired to terminals T5, T6 and T7 of the SM- I/O Lite module. The position of these switches can be observed at #15.04,5 and 6 respectively. These were assigned to the Encoder function block.

(* Feed encoder with inputs to be monitored for Blinking Fault/Status indicator *)



| T5 | T6 | T7 | # of Flashes | Comment |
|----|----|----|--------------|--------------------------------|
| 0 | 0 | 0 | 0 | No External Conditions exist |
| 0 | 0 | 1 | 1 | Only 1 condition on T7 |
| 0 | 1 | 0 | 2 | Only 1 condition on T6 |
| 0 | 1 | 1 | 3 | Conditions on T6 & T7 |
| 1 | 0 | 0 | 4 | Only 1 condition on T5 |
| 1 | 0 | 1 | 5 | A condition on T5 & T7 |
| 1 | 1 | 0 | 6 | Conditions on T5 & T6 |
| 1 | 1 | 1 | 7 | Conditions exist on all inputs |

NOTE

The SyPT Lite code for this example can be downloaded by clicking the following link. It should run on our Commander SK Training unit and flash the B3 indicator LED if parameter #8.41 is set to User (then RESET) and then #8.21 set to #18.50. (don't forget to depress RED reset to make these assignments take effect)

To obtain this SyPT Lite code example click here:

[CTSL008.dpl](#)

Summary

SyPT Lite can enable our drives to create a wide variety of useful functions at little or no cost that can enhance a product or machine operation.

Disclaimer

The SyPT Lite examples provided here are just that, **examples**. They can be used for reference when creating your own application solutions. Control Techniques does not warrant these examples "as is" for actual use. The examples are intended to stimulate ideas and facilitate application solutions. One always needs to consider and test **all aspects** of a system implementation to insure integrity and safety of their particular intended application.

Questions ?? Ask the Author:

Author: **Ray McGranor**
(716)-774-0093

e-mail : ray.mcgranor@emersonct.com